高速网络线速深度分组检测研究

徐克付 李阳 谭建龙 郭莉

摘要 网络的功能正逐步从简单的浅层分组处理向基于深度分组处理的复杂应用演化，适合高速核心网络（10Gbps）的线速深度分组检测技术成为了一个研究热点。本文指出了深度分组检测系统存在的“速度与性能之间的矛盾”及“语义失真”两个关键科学问题。围绕这两个问题，系统综述和深入分析了检测方法、检测模型、检测算法等方面出现的多种优化技术和策略，探讨了相关的研究动态与发展趋势，并总结了进一步的研究方向。

关键词 深度分组检测；特征集；模式匹配算法；性能评估

1 引言

深度分组检测（Deep Packet Inspection，DPI），有时也称为完全分组检测（Completely Packet Inspection），通过分析一系列分组的头部尤其是负载内容提供应用层语义感知的功能，属于应用级语义检测，是保障网络信息安全的基本手段与关键技术之一。

对于网络服务提供商，深度分组检测可以检测 OSI[1]参考模型的第二至第七层，从而使一系列新的网络安全功能成为可能，例如合法拦截、策略定义与保证、版权保护、内容审查等等。对于企业界，深度分组检测已成为网络信息安全的核心理论与关键技术，例如：网络入侵检测/保护系统（IDS/IPS）需要检测报文净负荷是否含有恶意计算机病毒；安全路由器转发分组前需要确定分组是否含有网络蠕虫代码；垃圾邮件（SPAM）过滤程序需要扫描报文里的垃圾邮件信息。除此之外，深度分组检测还允许细粒度控制，能够有效地防止缓冲区溢出攻击、DoS[2]攻击、经验丰富的黑客入侵等等。对于政府部门，深度分组检测作为管理信息传播的重要手段，使政府部门可以实现网络敏感内容的调查、网络舆情分析与传播、网络流量的监测与审查，对维护社会安全稳定，促进国民经济发展，乃至加强国防建设具有重大意义。

本文的研究工作是在国家 973 重点基础研究发展计划课题及国家自然科学基金项目“网络环境信息感知的线速深度分组检测研究”的支持下进行的。本文以动态变化的网络环境为研究对象，分析连接层流量环境、应用层语义环境等诸多网络环境因素对深度分组检测性能的影响，从深度分组检测机理上（检测方法-检测模型-检测算法）深入比较和分析现有的深度分组检测相关理论基础、各种优化技术及策略，并探讨相关的研究动态与发展趋势。

2 国内外相关研究

近年来，网络带宽和流量急剧增加，众多新型网络协议与应用不断涌现，对深度分组检测技术提出了更高的实时性要求以及更复杂的语义支持要求。下面从深度分组检测模型、检

[1] open system interconnect，国际标准化组织 (ISO) 的开放系统互连参考模型
[2] Deny of Service，拒绝服务

测算法、网络环境信息等方面作一评述。

2.1 深度分组检测模型

总体上，现有的检测模型缺乏准确性、灵活性和可扩展性，对更复杂的语义支持不够，同时深度分组检测系统的线速问题仍然没有解决。深度分组检测系统是保证网络空间更加洁净、网络空间中的国家主权更加完整的核心技术之一，涉及到国家机密、公民言论自由等敏感问题，得到各国的高度重视，目前已知的典型研究计划有以下几类：

(1) 可扩展性：

典型的有由美国 NSF(国家自然科学基金)提供支持的 STREAM 项目。其主要研究目标是研究一个通用的数据流管理系统，包括提供一个通用的、灵活的体系结构，相关的理论结果和算法、数据模型，相关的语言和语义，并探讨多个连续、快速、可变的网络流的查询与处理。目前他们的研究目标主要集中在可扩展性和性能优化方面。

(2) 语义支持：

典型的有也是由美国国家自然科学基金支持的 NIAGARA 项目，主要研究目标是在互联网环境下的 XML[3]数据检索和过滤系统。该系统在互联网上采集和监管信息，然后包装为 XML 数据流供检索和过滤使用。这样利用 XML 的语义信息，可以提供更加准确的数据流检索和过滤。

(3) 自适应：

典型的有加州大学伯克利分校的 Telegraph 项目。研究目标是对网络监听器的输出数据和 web 数据等提供自适应的查询。项目的特色是数据流的自适应查询处理，包括自适应连接和自适应操作调整。另外麻省理工学院和布朗大学的合作项目－Aurora，目标也是对各种各样的嵌入式设备产生的数据流进行监测。康奈尔大学也有个对传感器的数据进行查询和监管的项目 Cougar。

(4) 线速处理：

网络设备的线速处理，是目前国内外学术界和工业界同时热烈关注的问题。所谓线速处理，是指一个处理设备的输入数据，经过处理后可以没有延迟地传输到输出端（在排空完成的情况下），即可以以“在线传输速率”进行处理。这种情况在实际中是几乎不可能的，因为任何处理都需要时间。实际中的线速处理，一般是指输入的延迟控制在一定的范围内，比如说是输入速度的 20%以内的延迟。线速处理方法的有效性主要与两个因素密切相关：处理的复杂度和输入的速度。对复杂的处理，实现线速处理显然要困难得多；而对同样复杂度的处理，显然输入速度越高，线速处理将越困难。到目前为止，只有有限的论文涉及了线速处理的系统结构问题，特别是对于网络环境相关的信息流线速处理问题涉及更少。

2.2 深度分组检测算法

深度分组检测算法迄今为止已有数十种，它们的原理各不相同。传统的检测算法通常分为两个阶段：第一阶段是预处理过程，针对给定的模式集合 $\boldsymbol{P}$，根据不同的算法原理，使用不同的数据结构对全部模式进行预处理，生成特定的检测数据结构（泛称“检测自动机”）；第二阶段是搜索过程，对给定的文本 $\boldsymbol{T}$ 进行搜索，找出全部的结果。目前检测算法仍然是系统的速度瓶颈。这有两个方面的原因，分别产生于上面两个阶段：

[3] Extensible Markup Language，可扩展标记语言

(1) 检测算法不能适应变化着的网络数据流

检测算法的核心－检测自动机是在预处理阶段生成的，它仅仅通过对模式集合自身的处理来完成，没有考虑算法运行的网络环境信息。这带来的问题是：在不同的条件下，检测算法的性能表现千差万别，理论分析结果和实际运行效果不相符合。

在先前的研究中，一直把检测数据默认为是静态随机的，即与模式集合一样，是"等概率均匀分布"的。因此，认为检测数据不是算法给定的条件，理论上不需要对它进行考虑。基于这个前提，研究者给出了单模式匹配算法时间复杂度下限：$\Omega(n/m\log_{|\Sigma|}m)$，多模式匹配算法的时间复杂度下限：$\Omega(n/m\log_{|\Sigma|}mr)$。很多算法都已经达到或者接近这个最优值。

然而，网络数据流是应用过程中的输入，具有天然的动态性。在检测算法的研究中，考虑待检测数据的特性是最近几年才有的工作。文献[1]分析了检测算法与随机数据熵之间的关系，给出了在界定熵文本上的检测算法，并证明了其平均时间复杂度以及最坏情况下的时间复杂度。该研究结果得出结论：传统的检测数据"等概率均匀分布"的假设前提是不合理的。然而，该工作虽然考虑了模式的概率分布，随机数据的概率分布，但没有结合实际应用环境考虑它们之间的关系。动态的应用数据流对检测算法的性能影响还需要进一步研究。

(2) 检测算法在检测过程中的状态空间巨大

在影响检测算法性能的诸多因素中，存储空间大小占据着越来越重要的地位。这是因为，自动机是模式检测算法的主要数据结构，随着模式集合规模的增大，所需要的巨大存储空间导致缓存局部性变得很差，从而降低了算法的检测速度。因此，如何减少模式检测算法的存储空间、优化缓存的局部性，也是近年来研究的一个方向。

文献[2]提出一种分片的算法，选择出导致确定有限自动机（Deterministic Finite Automaton，DFA）状态膨胀的片段并将其隔离，从而降低了单个正则表达式存储需求。同时，文献基于正则表达式的组合关系提出一种选择性分组算法，在可以接受的存储需求总量下，通过选择性分群大幅度减少了状态机的个数，有效地降低了匹配算法的复杂性。文献[3]使用了一种基于 Bloom filter（布隆滤波器）的数据结构，将多条规则映射到一维向量空间上，有效地降低了存贮空间要求。文献[4]则是使用了对模式中的字符出现频率的统计，设计新的检测自动机结构，将经典的 AC[4]算法数据结构空间占用降低了 75%至 84%。文献[5]研究了经典检测算法的性能与缓存的关系，并针对不同类型的算法提出了相应的缓存优化策略，其试验结果表明，自动机的存储空间可压缩到不超过原始存储空间的 5%，速度上能够获得 2－3 倍的提升。

分析上述两个阶段，我们考虑的问题是：自动机应该在检测阶段而不是预处理阶段，随网络流量的变化，根据输入的数据流而自适应地创建或增减状态。这样自动机既能适应动态的网络数据流，又极大地减少空间要求，从而消除算法的速度瓶颈。

2.3 网络环境信息

相关研究的发展动态与趋势表明，流量环境信息与应用层语义环境信息，无论是对深度分组检测的速度、还是准确性，都有着显著的影响。

(1) 流量环境信息（提高检测速度）

文献[6]研究表明：网络流量特征的出现频率分布呈现不均匀性，少数流量特征出现频

[4] Aho-Corasick，阿霍-克若思克

率非常高。文献[7]的研究指出：对占据大部分网络流量的网络流进行识别和特别对待、处理有利于优化资源利用。文献[8]和[9]相关研究表明很大比例的网络流量承载的网络信息内容是相同或相似的。我们在前期的研究[10、11]中针对我国高速骨干网络流量，对应用层协议分布情况进行了统计，发现应用层协议类型分布是极其不均匀的，同时应用层数据内容分布也出现不均衡性。流行的、广泛应用的内容被大量用户点击，承载这些内容的流量在网络流量中占有很大比例。我们对观察到的数据进行分析，发现网络流量传输的内容偏斜和集中于一些当前流行的、常用的文件，包括：Web 网页、音乐文件、对等传输（P2P）共享文件、蠕虫病毒和垃圾邮件等。因此，传统的基于静态随机数据的深度分组检测模型与检测算法在理论设计与性能分析上已经不能适应动态变化的网络流量。

文献[6]利用文献[12]对网络流量属性特征的研究，提出了对深度分组检测的优化方法。文献[6]提出了一种专门的编码技术，用于建立自适应流量统计的哈夫曼（Huffman）树[5]，该方法可以采用并行处理，而且其最坏的情况有下限。文献[13]使用统计搜素树以缓存的形式来动态建立新的规则。这些新的规则具有更高的命中概率，支持根据预测网络流量的变化来自适应调整——在搜索树中，流量特性反应在树遍历过程中的不均衡性。大量搜索遍历的最终结果总是集中在少数几个叶子节点上。文献[14]研究了自适应流量的包过滤，该方法是一种实时包分类机制，取得了较好的效果。

我们在文献[15、16]中针对我国高速骨干网络流量，对应用协议分布情况进行了统计，发现应用协议类型分布是极其不均匀的，尽管有 100 多种应用层协议，但是使用最多的前 10 种协占据了超过 95%的网络流量，如表一所示。同时，正如前文所述，应用层数据内容分布也出现不均衡性。传统方法已经不能适应动态变化的网络流量。

表 1. 我国核心网 2008 年 3 月协议分布统计

协议名称	*HTTP*	*Bitorrent*	*eDonkey*	*SMTP*
协议比例	42.83%	31.01%	13.08%	4.72%
协议名称	*FTP*	*TSP*	*Gnutella*	*others*
协议比例	2.47%	01.60%	0.8%	03.5%

(2) 应用层语义环境[6]信息（提高检测准确度）

文献[17]指出，利用与上下文相关的语义信息的要求使得在检测数据中寻找特定模式的传统模式匹配方法已不能满足要求。深度分组检测程序依赖于应用层语义来做出有效的决策，需要更多的研究以建立可扩展的、高效的模式匹配方案。同时，越来越多的网络安全系统依赖于协议分析并从网络数据流中提取应用层的语义环境信息。

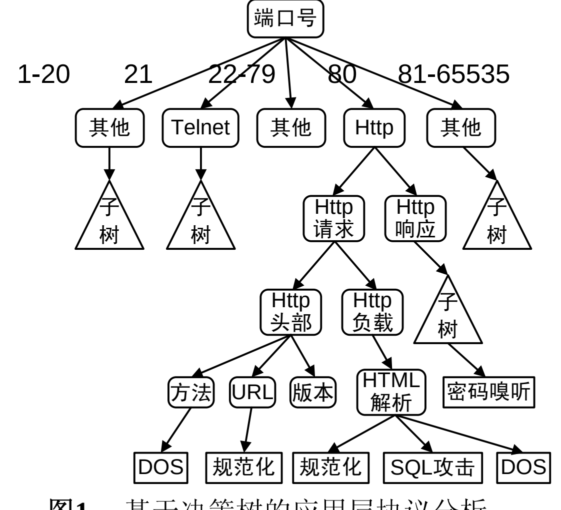


图1. 基于决策树的应用层协议分析

同时，应用层协议分析日渐成为深度分组检测的重要组成部分。文献[18]提出了一个基于网络动态应用层协议分析的入侵检测系统。对于每一个连接，该方法首先确定可能的应用层协议种类，然后使用适当的分析器分析协议并且提取应用层语义进行

[5] 又称最优二叉树

[6] 这里的语义，不是指自然语言理解的语义，而是指通信协议三要素（语义、语法与定时）中的语义，即与通信上下文相关的信息。

决策，取得了较好的效果。文献[19]提出了一个通用应用层协议分析（GAPA）架构及原型，GAPA 能够快速对协议进行在线分析，其工作非常值得借鉴，但是在自适应性、灵活性、决策的准确性等方面稍显不足。

文献[17]认为在深度分组检测模式匹配算法中，理解复杂的协议和应用层语义是非常有益的，有助于解释匹配成功的意义。调查显示，有近百分之八十的攻击来自于应用层[20]。应对这些攻击仅仅进行数据语法层面的检测是无能为力的，必需在应用层分析数据的内容及其通信的目的与意图。文献[21]通过使用决策树技术实施应用协议分析来提高检测能力，如图 1 所示。这种协议分析提取协议的特定部分，可大大减小特征搜索的空间。同时，该文献研究指出，基于应用层语义的处理方法提供了高层次的抽象，是提高检测准确性的合适选择。

3 关键科学问题与技术难点

3.1 关键科学问题

由于网络带宽和流量的急剧增加和众多新型协议与应用的不断涌现，深度分组检测必须支持更高的实时性要求以及更复杂的语义支持要求。“速度与性能之间的矛盾”及“语义失真”是当前深度分组检测系统的两个关键科学问题与技术难点。

深度分组检测的目标是深入检测主机之间交互的本质，并根据应用层语义进行决策。然而，现有的深度分组检测模型与算法无论是在理论设计上，还是在性能分析上，其假设的检测对象为语法层面的静态随机数据。而在实际环境中，在高速的网络流量中，检测的对象－网络数据时时刻刻在发生变化，网络流量的特性也不断变化，各种新型的网络协议层出不穷，应用层的语义环境也日益复杂多样。静态随机数据既不能反映真实网络流量的特性及其变化，也无助于理解复杂的协议和应用层语义，更不能解释检测成功的意义，往往造成严重的“语义失真”。

同时，由于网络信息的急剧膨胀，对通过深度分组检测的网络信息进行的复杂处理会使信息的输出有很大的延迟，使网络传输速度降低，带宽严重损失。这种延迟量随处理复杂度的增长一般是非线性函数。而对于高速的网络，如 IPv6，ATM 和千兆以太网，必须对高速到达的网络分组进行无延迟的线速（On-line Speed）检测。如果检测速度跟不上网络数据的传输速度，将会导致严重的拥塞，检测系统就会漏检和错检其中的部分数据，从而导致漏报、错报而影响系统的准确性和有效性。

3.2 技术难点

下面从深度分组检测机理上（检测方法-检测模型-检测算法）深入比较和分析现有的与深度分组检测相关的理论基础、各种优化技术及策略。

3.2.1 深度分组检测检测方法

3.2.1.1 特征集

“特征”是深度分组检测的关键概念。在法律上，指纹被用来鉴别公民（或个人）是否卷入犯罪事件，在深度分组检测中，特征（签名，Signature）被用来鉴别应用程序或恶意行为。

为了对成千上万的网络应用程序进行分析，必需提供有条理、系统化的鉴别方法。广义而言，特征就是模式碎片。这些模式碎片被筛选出来，应该可以独一无二地鉴别与之关联的

应用程序（如图 2 所示）。例如：应用协议使用特别的信息头初始化并控制信息的传递；网络入侵、蠕虫、病毒含有特定代码片断；待查询的特定信息也可视为特征。一般将这些特征表示成正则表达式的形式。当发现新的应用程序或恶意行为时，就对其进行分析，设计出合适的特征并添加到数据库中。此数据库一般称之为特征库。限于篇幅，本文对分析发现新特征的方法和技术不加评述，具体可参见文献[22，23]。

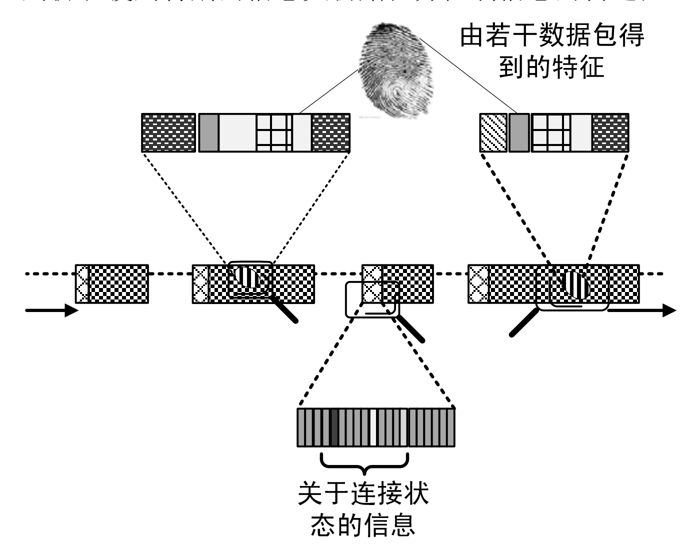


图2. 深度分组检测根据特征集鉴别会话

特征集的复杂性及其描述方法，对检测速度及检测的准确性产生了严重影响。导致深度分组检测系统难以线速化，影响检测准确性与实用性的主要是特征集以下四方面特点：

(1) **网络协议多种多样**，各种新型的协议不断涌现，各种新的安全攻击也层出不穷，导致特征集数量巨大。比如 Snort 入侵检测系统截止到 2006 年四月就已包含 4867 条规则，而且每条规则里面包含多个特征。另外，网络环境日新月异，导致特征集不断变化，不断更新，具有动态性。

(2) .**特征存在重复现象**，每个分组可能与多个攻击相关，比如在 Snort 系统中，一个 HTTP 分组可能存在 1096 个攻击漏洞。

(3) **特征在分组中的出现位置具有不确定性**，因为应用程序的格式复杂多变，通常不能准确地确定特征在分组中的出现位置，因此必须高速检查分组头部及负载的每一字节。

(4) 正则表达式可以表述比单字符串、字符串集合、扩展字符串更复杂的模式，由于其强描述能力和灵活性，在深度分组检测系统得到了广泛的应用。著名的开源系统 Snort、Bro 等，均采用正则表达式形式的特征集。然而**正则表达式与应用协议的语义之间存在着鸿沟**，仅仅根据正则表达式匹配的结果，难以进行应用语义级决策。

3.2.1.2 特征碎片与分组乱序

深度分组检测中寻找恶意模式一般是根据预先定义的特征集，对分组任意位置的内容使用正则表达式匹配算法进行精确模式匹配。在一个网络通信会话里，通信的数据被拆分成多个部分分别封装在多个不同的分组中，待检测的目标模式往往跨越多个分组，因此以单个分组为检测单位无法检测出整个模式。

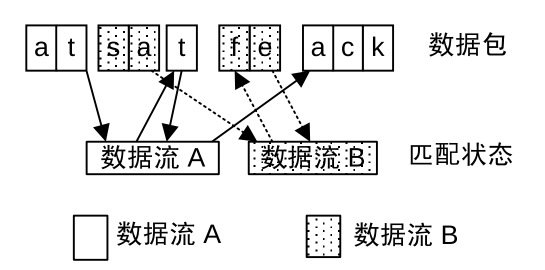


图3. 分组级检测无法检测出整个模式

如图 3 所示，对单个分组进行检测无法检测出模式“attack”，因为它被拆分地分布在三

个不同的分组里。对于需要完全检测的安全类应用程序，比如 NIDS[7]，这个问题非常严重。因为攻击者往往故意拆分攻击模式以规避检测，为了发现这些模式碎片，必须进行会话级的检测。

深度分组检测的另一个重要问题是分组的乱序处理与会话重组[24]。由于网络环境的复杂性，对于模式匹配单元，属于同一个会话的分组物理到达顺序往往并不是逻辑连续的，称之为乱序（Out of Order）。后续分组乱序时，模式匹配过程将会终止，如图 4 所示。

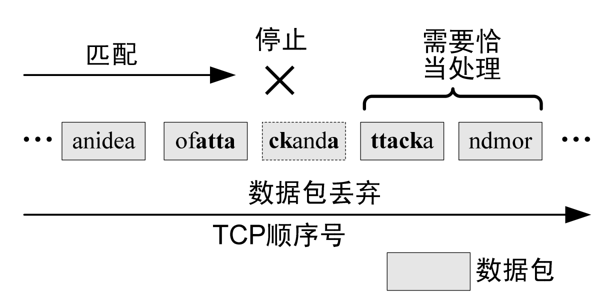


图4. 分组乱序导致匹配终止

对分组乱序问题已经有人在不同的网络环境下分别做了相关的研究。文献[25]采用了发送 ICMP[8]分组并分析其响应的方法。这种方法的主要问题是不同的网络对待 ICMP 流量处理的方法各不相同。文献[26]采用不同的方法研究了端到端的 TCP[9]会话；文献[27]进行了更广泛的分析，采用了在网络中设置单个捕获点来检测乱序分组和 TCP 会话；文献[28]采用了与[27]相似的方法分析 TCP 会话的吞吐量，并重点研究了 TCP 重组窗口对吞吐量及通信性能的影响；文献[29]研究了 UDP[10]会话的重组问题，使用的方法是基于流的重组，其主要工作放在了流内部的重组机制方面，并对突发流量进行了相关研究。

为了获得精确的匹配结果，后续分组需要做适当的处理，处理方法一般有缓冲后续乱序分组和丢弃后续乱序分组两种。

缓冲后续乱序分组的方法[29, 30]将后续的乱序分组存储在缓冲区中，根据分组的顺序号重新进行排序，把排好序的分组依次送入模式匹配单元进行检测。当所需的下一个逻辑分组未到达时，检测单元停止并等待后续逻辑连续的分组到达。缓冲后续乱序分组需要的缓冲空间为[往返时间 (RTT) [11]] ×[网络带宽]，假设 RTT=200ms，带宽=40Gbps，所需的存储空间达到 1GB。在最坏的情况下，缓冲高速到达的乱序分组会使所需的存储空间持续增加。由于网络高速增长且增速高于存储容量，在可以预见的未来此问题将变得更为严重。

另一种方法是丢弃后续所有逻辑顺序不一致的分组[31]，由于 TCP 协议的延时重传机制，模式匹配单元会最终获得逻辑一致的后续分组。这种方法的弊端是由于大量的分组被丢弃，TCP 协议的流控制机制会使网络的吞量急剧下降。在 TCP 协议中，由分组丢失带来的恢复时间是 RTT×分组丢失数量。而且，如果延时重传时间 RTO（retransmission time out）超出，汇聚窗口(CWND)将被重置到最小。

3.3 深度分组检测模型

3.3.1 建模

这里我们以应用协议分类为例对深度分组检测进行建模与性能评估，其他两类深度分组检测与本例相同或类似。深度分组检测协议分类模型的结构如图 5 所示。通过观察，每个应

[7] Network Intrusion Detection System，网络入侵检测系统
[8] Internet Control Message Protocol，因特网信息控制协议
[9] Transmission Control Protocol，传输控制协议
[10] User Datagram Protocol，用户数据报协议
[11] round trip time

用协议都会使用特别的信息头初始化并控制信息的传递。模型为两部分：慢速通道和快速通道。慢速通道定义未分类会话的处理方法及所使用的特定算法（深度分组检测使用正则表达式），快速通道根据慢速通道的结果把后续的分组（属于同一会话的）关联到正确的协议[32]。

分类器模型包括五个处理单元：会话 ID 抽取单元、会话查找单元、模式匹配单元、会话更新单元以及会话关联单元。协议分类器运行时，如果一个会话的分组尚无法判别属于哪种协议，就把该分组传递给深度分组检测分类器。分类器使用协议特征集对该分组进行模式匹配。一旦某个特征匹配成功，就将该分组（和整个 TCP 会话）判别为相应的应用协议类别，属于该会话的其它后续分组则跳过特征匹配单元。当一个新的会话被成功鉴别后，就在会话表中创建一个新条目。这个条目包括 5 元组、应用协议类别和时间戳。时间戳用来记录分类器观察到该会话最后一个分组的时间。快速通道的主要功能是更新会话表中当前会话的时间戳，删除在会话表中不活动的会话，以防止会话结束后该会话没有从会话表中删除，这对 UDP 数据流尤为普遍，10 分钟的会话超时通常认为该会话已经结束[33]。

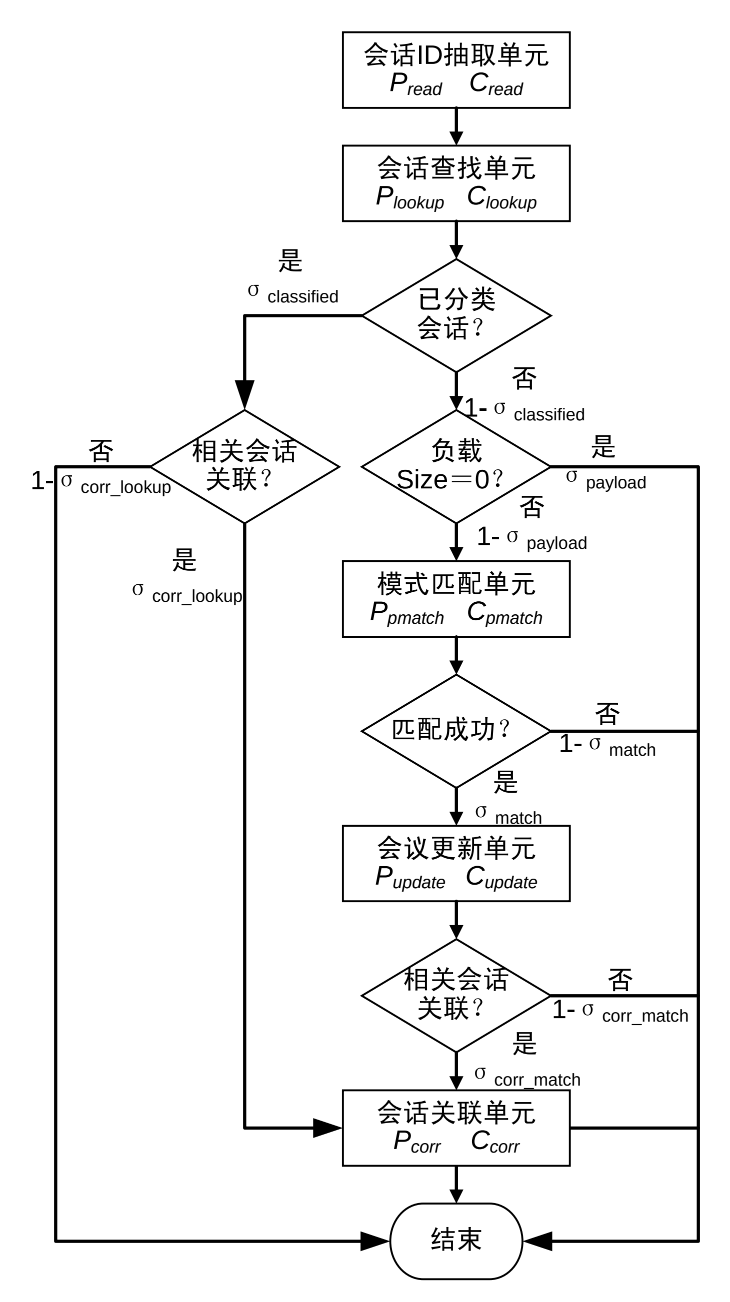


图5. 深度分组检测模型及性能评估

3.4 深度分组检测算法

根据自动机和文法理论，正则表达式具有与自动机等价的描述能力，因此可以通过构造正则表达式对应的自动机来识别正则表达式的语言。自动机分为两种形式：一种是非确定有限自动机（Nondeterministic Finite Automaton，NFA），另外一种是确定有限自动机（Deterministic Finite Automaton，DFA）。非确定有限自动机和确定有限自动机的区别在于：对于确定的状态和确定输入，非确定有限自动机允许有多个后继状态，而确定有限自动机则只能有唯一的后继状态。非确定有限自动机和确定有限自动机在描述能力上是等价的。

图 6 是用自动机进行正则表达式匹配的一般过程。首先，正则表达式被解析成解析树，然后再转换成非确定有限自动机。目前，从解析树构造非确定有限自动机的常用方法有汤普

逊（Thompson）构造法和柯罗希克夫（Glushkov）构造法。构造好非确定有限自动机后，可以直接用它进行文本匹配，也可以将其确定化和最小化来构造相应的确定有限自动机，然后用确定有限自动机进行文本匹配。目前有三种方法：

图6. 正则表达式搜索的一般过程

(1) 基于非确定有限自动机的匹配方法

前面已经介绍过，对于确定的状态和确定的输入，非确定有限自动机允许有多个到达状态。因此基于非确定有限自动机的匹配方法需要将当前的活动状态存储起来，对每次读入的新字符，依次查看当前的每一个活动状态，以获得被激活的新状态；然后将这些新状态加入到一个新的活动状态集合。

设正则表达式的长度为 m，则每个状态最多有 m 个可到达状态，并且当前最多可有 m 个活动状态。通过使用位向量的方法，基于非确定有限自动机的匹配方法的状态转换时间复杂度为 $O(m)$。当 m 比较大时，匹配速度比较慢，但其空间复杂度仅为 $O(m)$，并且单纯的非确定有限自动机匹配引擎易于实现。

(2) 基于确定有限自动机的匹配方法

基于确定有限自动机的匹配方法利用了确定有限自动机的特性：对于确定的状态的确定输入，确定有限自动机只能有唯一的到达状态。因此，该类方法的状态转换时间复杂度为 $O(1)$，匹配速度大大提高。但是与非确定有限自动机相比，确定有限自动机有可能带来空间上的指数增长，最坏情况下确定有限自动机的空间复杂度为 $O(2^m)$（m 为正则表达式的长度）。在当前的应用需求中，m 往往比较大，这是无法忍受的。

(3) 混合方法

针对非确定有限自动机较慢的匹配速度和确定有限自动机较大的存储空间，研究人员提出了一种折中的方法[34]，它介于非确定有限自动机和确定有限自动机之间，该方法的核心思想是：首先把非确定有限自动机划分成 k 个模块，然后分别对每个模块构建确定有限自动机。划分之后的非确定有限自动机相当于最多有 m/k （m 为正则表达式规模）个活动状态，每个模块最坏情况下需要 $O(2^k)$ 空间，因此该方法的状态转换时间复杂度为 $O(m/k)$，空间复杂度为 $O(m\times 2^k/k)$。该方法的一个缺点是对非确定有限自动机的划分，即决定如何选择合适的 k 值以及每个状态应该划分到哪个模块，比较困难。表 2 是上述三种方法的复杂度比较：

表 1. 基于 NFA 和 DFA 匹配的复杂度比较

匹配方法	状态转换时间复杂度	空间复杂度
NFA	$O(1)$	$O(m)$
DFA	$O(m)$	$O(2^m)$
混合方法	$O(m/k)$	$O(m\times 2^k/k)$

4 未来研究方向与展望

尽管近年来涌现了很多解决深度分组检测关键问题与技术难点的研究思路、方法和算

法，但是还存在许多需要进一步深入研究的问题：

(1) **如何进一步提高处理能力** 随着高速处理要求的日益增加，可以预计，今后深度分组检测的应用将会更多地以硬件实现。但为了兼顾灵活，也会相应地引入一些辅助的软件策略。目前，在普通硬件上，以 Snort 规则集作为模式集合的实现，其吞吐量一般不超过 l0G，还有可能通过改进辅助器件和优化策略来进一步提高。

(2) **如何快速检测加密了的数据** 加密了的数据一般不能直接被深度分组检测到。目前已经有了相应的方法，就是在深度分组检测组件中加入解密插件。然而解密算法耗时大，对线速检测影响很大。如何快速实现对加密了的数据进行深度检测，是需要进一步研究的问题。

(3) **如何实时检测无特征的攻击** 本文假设特征集是预先已知的，然而在某些情况下，比如蠕虫爆发后，及时散播新的特征以供检测并不是一件十分容易的事情。因为某些蠕虫爆，比如 Slammer worm devastation，发生时会侵占大部分网络带宽，引起网络拥塞。因此需要新的算法在没有特征集的条件下检测可疑的流量，并在得到新的特征前限制可疑流量的速率。

参考文献：

[1] Shenfeng Chen, J.H. Reif, *"Fast pattern matching for entropy bounded text," DCC, pp：282－288, Data Compression Conference* (DCC07), 2007

[2] 徐 乾，鄂跃鹏，葛敬国等. *深度包检测中一种高效的正则表达式压缩算法. 软件学报[J]. Vol.20, No.8*, August 2009, pp.214－226

[3] 叶明江，崔勇，徐恪等. *基于有状态 Bloom filter 引擎的高速分组检测；软件学报[J]. Vol.18，No.1*，January 2007 .pp：117－126

[4] YU Jianming, XUE Yibo, LI Jun. *Memory Efficient String Matching Algorithm for Network Intrusion Management System, TSinghua Science and Technology*, 2007，12(5).

[5] T. Jian-long, L. Yan-bing, L. Ping. Accelerating Multiple String Matching By Using Cache-efficient Strategy. *The Ninth International Conference on Web-Age Information Management (WAIM)*, 2008

[6] A. El-Atawy, T. Samak, E. Al-Shaer, and H. Li. *On using online traffic statistical matching for optimizing packet filtering performance. In IEEE INFOCOM'07*, May 2007.

[7] E. Cohen,C. Lund. *Packet classification in large ISPs:design and evaluation of decision tree classifiers. In Proceedings of SIGMETRICS, Banff, Canada*, 2005.

[8] A. Madhukar and C. Williamson. *"A Longitudinal Study of P2P Traffic Classification"*. MASCOTS 2006.

[9] Ma, Levchenko, Kreibich, Savage, and Voelker. *"Unexpected means of protocol inference". Internet Measurement Confererence*, 2006.

[10] [10]Xu Kefu, Fang Binxing，Guo Li, Tan Jianlong. *Traffic-Aware Frequent Elements Matching Algorithms for Deep Packet Inspection*. NSWCTC2010 已录用

[11] Tingwen Liu、Yifu Yang、Yong Sun、Li Guo . *Fast and Memory-Efficient Traffic Classification with Deep Packet Inspection in CMP Architecture*.已录用

[12] H. Hamed, A. El-Atawy, and E. Al-Shaer. *Adaptive statistical optimization techniques for firewall packet filtering. In IEEE INFOCOM'06*, April 2006.

[13] Qunfeng Dong, Suman Banerjee, Jia Wang, and Dheeraj Agrawal. *Wire speed packet classification without tcams: a few more registers (and a bit of logic) are enough. SIGMETRICS Perform. Eval.*

Rev., 35(1):253– 264, 2007.

[14] Kencl L1Schwarzer C.*Traffic adaptive packet filtering of denial of service attacks[C]//Proceedings of the 2006 International Symposium on world of Wireless, Mobile and Multimedia Networks, Sashington*, 2006:485-489

[15] Xu Kefu，Guo Li，Tan Jianlong， etc. *Traffic-Aware Frequent Elements Matching Algorithms for Deep Packet Inspection[C]. Proc of the IEEE International Conference on Networks, Security，Wireless Communications and Trusted Computing.* Page(s): Vol（2）:93-96，2010.

[16] [16]Tingwen Liu, Yong Sun, Li Guo, "*Fast and Memory-Efficient Traffic Classification with Deep Packet Inspection in CMP Architecture[C]" nas，pp.208-217, 2010 Fifth International Conference on Networking, Architecture, and Storage*, 2010

[17] PC Lin, YD Lin, YC Lai, TH Lee . *Using string matching for deep packet inspection. Computer, 2008 - doi.ieeecomputersociety.org* 2008

[18] H Dreger, A Feldmann, M Mai, V Paxson, *Dynamic application-layer protocol analysis for network intrusion detection. 15th USENIX Security Symposium* Pp. 257–272. 2006.

[19] Vaidehi, V.; Srinivasan, N.; Anand, P.; Balaji, A.P.; Prashanth, V.; Sangeetha, S.A *Semantics Based Application Level Intrusion Detection System. International Conference on Signal Processing, Communications and Networking*, 2007. ICSCN '07. 22-24 Feb. 2007 Page(s):338 – 343

[20] www.lsi.com/documentation/working/tarari_content_processors/TarariRAX Whitepaper.pdf

[21] A.Anitha and V.Vaidehi. *Context based Application Level Intrusion Detection System[C]. International conference on Networking and Services,* 2006. ICNS '06. 16-18 July 2006 Page(s):16 - 20

[22] Alessandro Finamore, Marco Mellia，Michela Meo， etc. *KISS:Stochastic Packet Inspection Classifier for UDP Traffic[J]. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL.18(5)*,pp:1505-1515,2010

[23] 唐勇，卢锡城，王勇军.*攻击特征自动提取技术综述[J]，通信学报，Vol.3(2)*，pp：96-105，2009

[24] V. Paxson，*"End-to-end internet packet dynamics[C]" in IEEE/ACCM Transactions on Networking*，1999，pp. 277–292.

[25] S. Jaiswal，G. Iannaccone，C. Diot，J. Kurose，and D. Towsley，*"Measurement and Classification of Out-of-Sequence Packet in a Tier-1 IP Backbone[C]" in Internet Measurements Workshop (IMW)* ，2002.

[26] M. Laor and L. Gendel，*"The effect of packet reordering in a backbone link on application throughput[J]" in IEEE Network*，2002.

[27] X. Zhou and P. V. Mieghem，*"Reordering of IP Packets in Internet[C]" in C. Barakat，I. Pratt (Eds.)：Fifth annual Passive and Active Measurement Workshop PAM2004，LNCS 3015*，pp. 207-218，2005. Springer-Verlag Berlin Heidelberg 2004.

[28] M. Necker，D. Contis，D. Schimmel，*"Tcp-stream reassembly and state tracking in hardware[C]" in Proc. of 10 the Annual IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM'02)，September 2002*，pp. 286–287.

[29] S. Li, J. Tørresen，and O. Sorasen，*"Exploiting Stateful Inspection of Network Security in Reconfigurable Hardware[C]" in Proc. of 13th Int. Conf. on Field Programmable Logic and Applications(FPL '03)，September 2003*，pp. 1153–1157.

[30] D. V. Schuehler，J. Moscola，J. Lockwood，*"Architecture for Hardware Based TCP/IP Content Scanning System[C]" in Proc. of 11th IEEE Symp. on High Performance Interconnects，August 2003*，pp：89 – 94.

[31] M. Fish and G. Varghese，*"Fast Content-Based Packet Handling for Intrusion Detection[R]" UCSD technical report CS2001-0670*

[32] Niccol Cascarano, Alice Este, Francesco Gringoli etc. *An Experimental Evaluation of the Computational Cost of a DPI Traffic Classifier[C]. Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, Honolulu, HI* ,page(s): 1 - 8

[33] N. Brownlee，*Traffic flow measurement: Meter MIB, Request for Comments RFC 2064, Internet Engineering Task Force*，January 1997.

[34] 徐 乾，鄂跃鹏，葛敬国等. *DPI 中一种高效的正则表达式压缩算法. 软件学报[J]. Vol.20, No.8*, August 2009, pp.214－226

作者简介：

徐克付： 中科院计算所助理研究员 xukefu@software.ict.ac.cn

李　阳： 中科院计算所博士研究生

谭建龙： 中科院计算所副研究员

郭　莉： 中科院计算所信息安全研究中心主任、正研级高级工程师